Step 1: Install Apache2 HTTP Server on Ubuntu

Apache2 HTTP Server represents the A in the LAMP stack… It's the most popular web server in use… so install it, since Joomla needs it..
To install Apache2 HTTP on Ubuntu server, run the commands below…

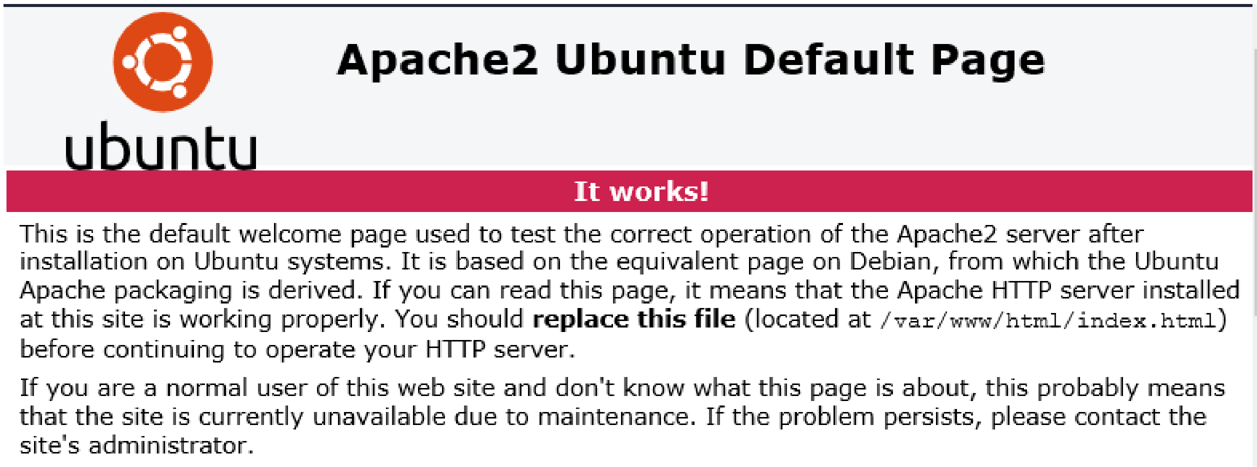
```
sudo apt update
sudo apt install apache2
```

After installing Apache2, the commands below can be used to stop, start and enable Apache2 service to always start up with the server boots.

```
sudo systemctl stop apache2.service
sudo systemctl start apache2.service
sudo systemctl enable apache2.service
```

Test Apache, you open web browse, on url you write: http://ipserver or http://hostname

You look below:



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Step 2: Install MariaDB Database Server

MariaDB database server is a great place to start when looking at open source database servers to use with Joomla… To install MariaDB run the commands below…

```
sudo apt-get install mariadb-server mariadb-client
```

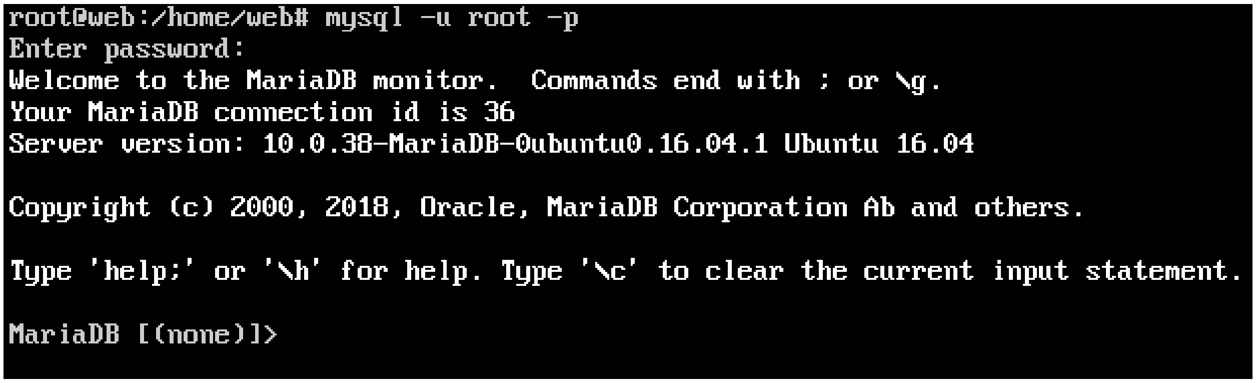
After installing MariaDB, the commands below can be used to stop, start and enable MariaDB service to always start up when the server boots..

Run these on Ubuntu 16.04 LTS

```
sudo systemctl stop mysql.service
sudo systemctl start mysql.service
sudo systemctl enable mysql.service
```

To test if MariaDB is installed, type the commands below to logon to MariaDB server

```
sudo mysql -u root -p
```



```
root@web:/home/web# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.0.38-MariaDB-0ubuntu0.16.04.1 Ubuntu 16.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

Step 3: Install PHP 7.2 and Related Modules

Run the commands below to add the below third party repository to upgrade to PHP 7.2

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:ondrej/php
```

Then update and upgrade to PHP 7.2

```
sudo apt update
```

Next, run the commands below to install PHP 7.2 and related modules.

```
sudo apt install php7.2 libapache2-mod-php7.2 php7.2-common php7.2-mbstring php7.2-xmlrpc
php7.2-soap php7.2-gd php7.2-xml php7.2-intl php7.2-mysql php7.2-cli php7.2-zip php7.2-curl
```

After installing PHP 7.2, run the commands below to open PHP default config file for Apache2…

```
sudo nano /etc/php/7.2/apache2/php.ini
```

Then make the changes on the following lines below in the file and save. The value below are great settings to apply in your environments.

```
file_uploads = On
allow_url_fopen = On
memory_limit = 256M
upload_max_filesize = 100M
max_execution_time = 360
date.timezone = America/Chicago
```

Step 4: Restart Apache2

After installing PHP and related modules, all you have to do is restart Apache2 to reload PHP configurations…

To restart Apache2, run the commands below

```
sudo systemctl restart apache2.service
```

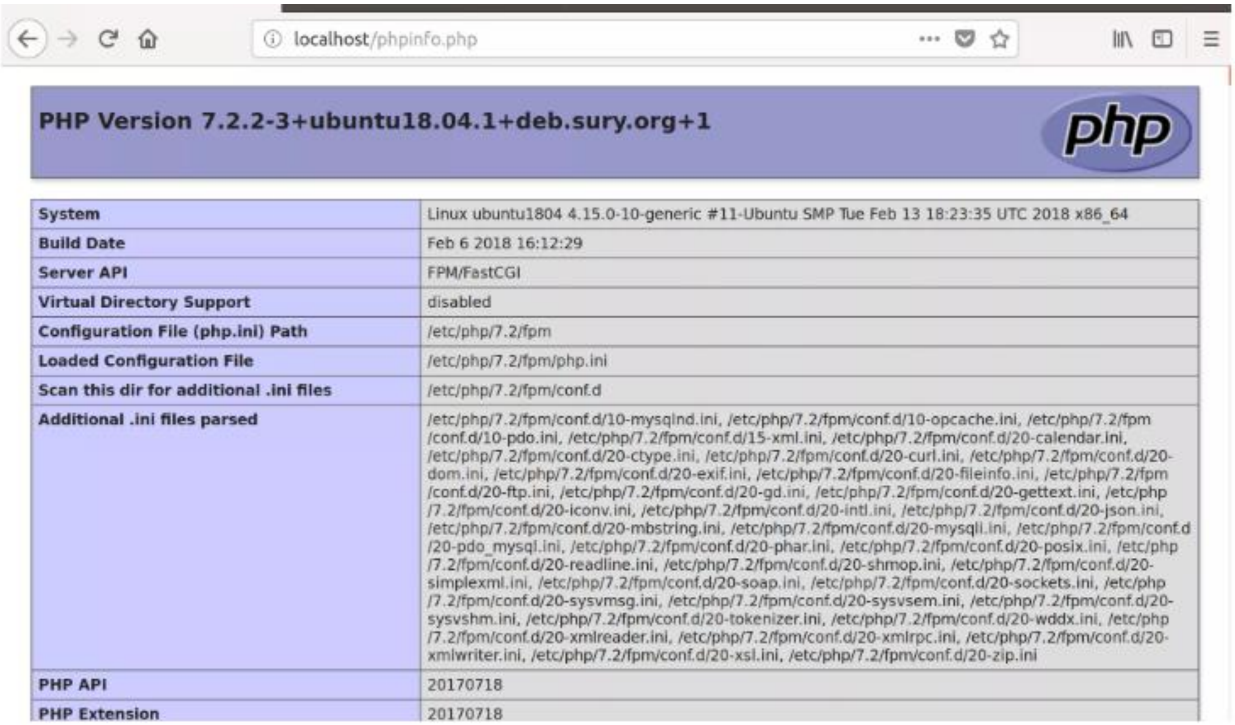
To test PHP 7.2 settings with Apache2, create a phpinfo.php file in Apache2 root directory by running the commands below

```
sudo nano /var/www/html/phpinfo.php
```

Then type the content below and save the file.

```
<?php phpinfo( ); ?>
```

Save the file.. then browse to your server hostname followed by /phpinfo.php http://localhost/phpinfo.php



PHP Version 7.2.2-3+ubuntu18.04.1+deb.sury.org+1

System	Linux ubuntu1804 4.15.0-10-generic #11-Ubuntu SMP Tue Feb 13 18:23:35 UTC 2018 x86_64
Build Date	Feb 6 2018 16:12:29
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/fpm
Loaded Configuration File	/etc/php/7.2/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/fpm/conf.d
Additional .ini files parsed	/etc/php/7.2/fpm/conf.d/10-mysqlnd.ini, /etc/php/7.2/fpm/conf.d/10-opcache.ini, /etc/php/7.2/fpm/conf.d/10-pdo.ini, /etc/php/7.2/fpm/conf.d/15-xml.ini, /etc/php/7.2/fpm/conf.d/20-calendar.ini, /etc/php/7.2/fpm/conf.d/20-ctype.ini, /etc/php/7.2/fpm/conf.d/20-curl.ini, /etc/php/7.2/fpm/conf.d/20-dom.ini, /etc/php/7.2/fpm/conf.d/20-exif.ini, /etc/php/7.2/fpm/conf.d/20-fileinfo.ini, /etc/php/7.2/fpm/conf.d/20-ftp.ini, /etc/php/7.2/fpm/conf.d/20-gd.ini, /etc/php/7.2/fpm/conf.d/20-gettext.ini, /etc/php/7.2/fpm/conf.d/20-iconv.ini, /etc/php/7.2/fpm/conf.d/20-intl.ini, /etc/php/7.2/fpm/conf.d/20-json.ini, /etc/php/7.2/fpm/conf.d/20-mbstring.ini, /etc/php/7.2/fpm/conf.d/20-mysqli.ini, /etc/php/7.2/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.2/fpm/conf.d/20-phar.ini, /etc/php/7.2/fpm/conf.d/20-posix.ini, /etc/php/7.2/fpm/conf.d/20-readline.ini, /etc/php/7.2/fpm/conf.d/20-shmop.ini, /etc/php/7.2/fpm/conf.d/20-simplexml.ini, /etc/php/7.2/fpm/conf.d/20-soap.ini, /etc/php/7.2/fpm/conf.d/20-sockets.ini, /etc/php/7.2/fpm/conf.d/20-sysvmsg.ini, /etc/php/7.2/fpm/conf.d/20-sysvsem.ini, /etc/php/7.2/fpm/conf.d/20-sysvshm.ini, /etc/php/7.2/fpm/conf.d/20-tokenizer.ini, /etc/php/7.2/fpm/conf.d/20-wddx.ini, /etc/php/7.2/fpm/conf.d/20-xmlreader.ini, /etc/php/7.2/fpm/conf.d/20-xmlrpc.ini, /etc/php/7.2/fpm/conf.d/20-xmlwriter.ini, /etc/php/7.2/fpm/conf.d/20-xsl.ini, /etc/php/7.2/fpm/conf.d/20-zip.ini
PHP API	20170718
PHP Extension	20170718

Step 5: Create Joomla Database

Now that you've installed all the packages that are required for Joomla to function, continue below to start configuring the servers. First run the commands below to create a blank Joomla database.
To logon to MariaDB database server, run the commands below.

```
sudo mysql -u root -p
```

Then create a database called joomladb

```
CREATE DATABASE joomladb;
```

Create a database user called joomlauser with new password

```
CREATE USER 'joomlauser'@'localhost' IDENTIFIED BY 'new_password_here';
```

Then grant the user full access to the database.

```
GRANT ALL ON joomladb.* TO 'joomlauser'@'localhost' IDENTIFIED BY 'user_password_here' WITH
GRANT OPTION;
```

Finally, save your changes and exit.

```
FLUSH PRIVILEGES;
EXIT;
```

Step 6: Download Joomla Latest Release

Next, visit Joomla site and download the latest package.... or run the commands below to download and extract Joomla content.

After downloading, run the commands below to extract the downloaded file and move it into a new Joomla root directory.
Link download:
https://downloads.joomla.org/cms/joomla3/3-7-0
After download move file **Joomla! 3.7.0** into server ubuntu by FileZilla

```
sudo apt-get install unzip
sudo mkdir -p /var/www/html/joomla
sudo unzip Joomla*.zip -d /var/www/html/joomla
```

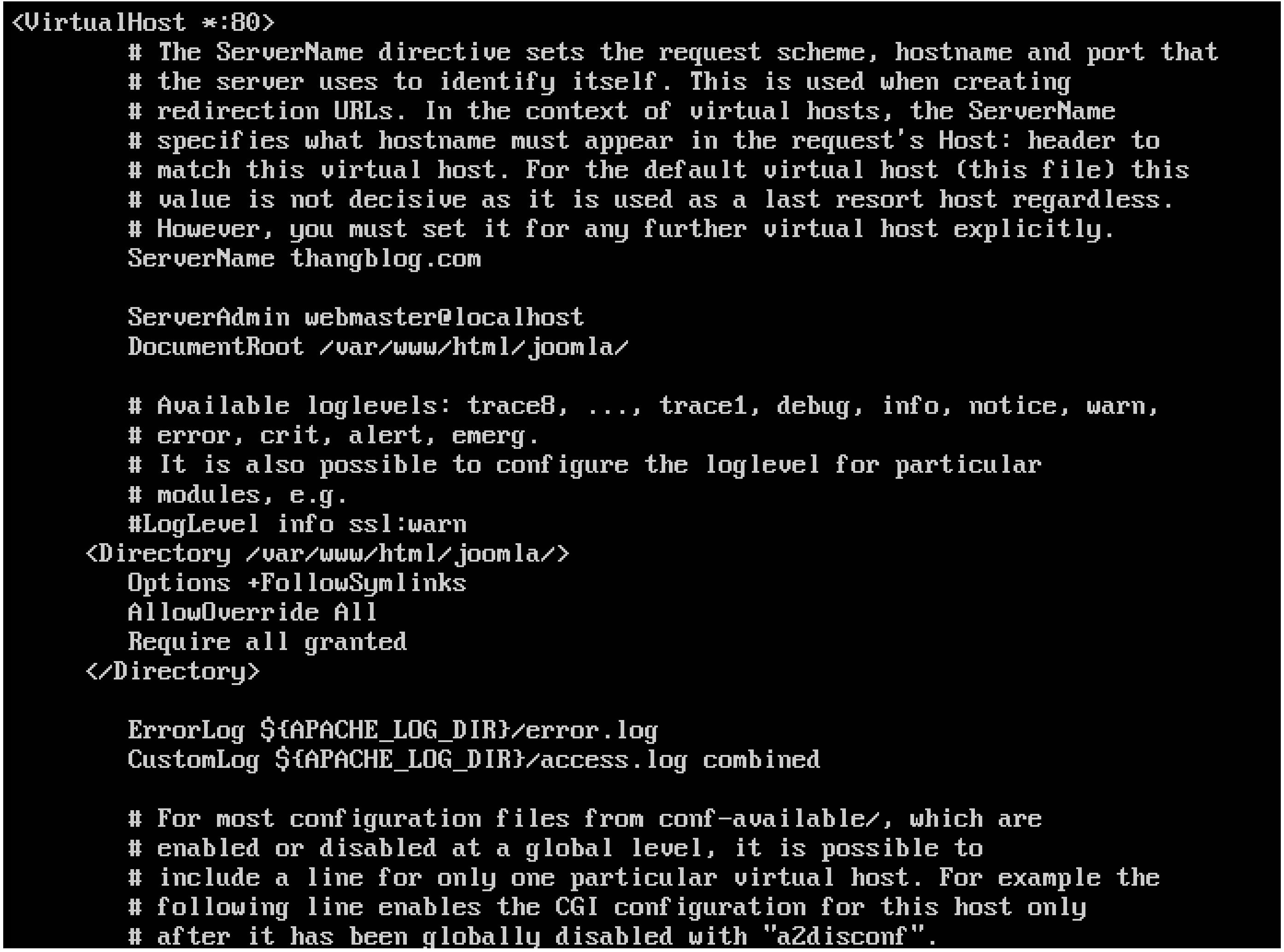
Then run the commands below to set the correct permissions for Joomla to function properly.

```
sudo chown -R www-data:www-data /var/www/html/joomla/
sudo chmod -R 755 /var/www/html/joomla/
```

Step 7: Configure Apache2 Joomla Site

Finally, configure Apache2 configuration file for Joomla. This file will control how users access Joomla content. Run the commands below to create a new configuration file called joomla.conf

`sudo nano /etc/apache2/sites-available/000-default.conf` below…



```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        ServerName thangblog.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/joomla/

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn
     <Directory /var/www/html/joomla/>
        Options +FollowSymlinks
        AllowOverride All
        Require all granted
     </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
```